Energy Characterization and Optimization of Image Sensing Toward Continuous Mobile Vision

Robert LiKamWa[†,‡], Bodhi Priyantha[‡], Matthai Philipose[‡], Lin Zhong[†,‡], and Paramvir Bahl[‡]

[†]Rice University, Houston, TX [‡]Microsoft Research, Redmond, WA

ABSTRACT

A major hurdle to frequently performing mobile computer vision tasks is the high power consumption of image sensing. In this work, we report the first publicly known experimental and analytical characterization of CMOS image sensors. We find that modern image sensors are not energy-proportional: energy per pixel is in fact inversely proportional to frame rate and resolution of image capture, and thus image sensor systems fail to provide an important principle of energy-aware system design: trading quality for energy efficiency.

We reveal two energy-proportional mechanisms, supported by current image sensors but unused by mobile systems: (*i*) using an optimal clock frequency reduces the power up to 50% or 30% for low-quality single frame (photo) and sequential frame (video) capturing, respectively; (*ii*) by entering low-power standby mode between frames, an image sensor achieves almost constant energy per pixel for video capture at low frame rates, resulting in an additional 40% power reduction. We also propose architectural modifications to the image sensor that would further improve operational efficiency. Finally, we use computer vision benchmarks to show the performance and efficiency tradeoffs that can be achieved with existing image sensors. For image registration, a key primitive for image mosaicking and depth estimation, we can achieve a 96% success rate at 3 FPS and 0.1 MP resolution. At these quality metrics, an optimal clock frequency reduces image sensor power consumption by 36% and aggressive standby mode reduces power consumption by 95%.

Categories and Subject Descriptors

I.4.m [**Image Processing and Computer Vision**]: Miscellaneous; I.5.4 [**Performance of Systems**]: Modeling techniques, Performance attributes

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Image sensor; energy efficiency; mobile systems; computer vision; energy proportionality

1 Introduction

Cameras are ubiquitous on mobile systems, from laptops, tablets, smartphones, to wearable devices, such as Google Project Glass or GoPro Helmet Cameras. Originally intended for capturing photo or video, cameras have inspired many to provide new mobile computer vision services, including marker-identification, gesture-based interaction, and object recognition. Many researchers, including ourselves [2], also envisage that by showing computers what we see on the go, we will see a new generation of personal computing coming, or *continuous mobile vision*. Unfortunately, image sensing, the very first stage of any vision-based application, is power-hungry, consuming hundreds of milliWatts. As a result, users and developers refrain from using the camera extensively. For example, most computer vision applications for smartphones are intended for occasional, instead of continuous, use; wearable cameras are designed for on-demand capture rather than continuous on-the-go capture.

Modern mobile systems employ CMOS image sensors [5] due to their low power and low cost. CMOS image sensors are an active area of circuit research where power consumption, image quality and cost of fabrication have been the main focuses of improvement. However, mobile systems integrate these image sensors with such a narrowly defined hardware and software interface that typically only the frame resolution and sometimes the frame rate can be changed in software. Furthermore, as we show later, reducing the image quality does not currently provide significant power reduction. The image sensor remains a black box to system and application developers with its system behavior, in particular power consumption, not well understood.

In this work, we provide a comprehensive treatment of the energy characteristics of image sensors in the context of computer vision applications. In particular, we consider (*i*) how the energy consumption of an image sensor is related to its image quality requirements, i.e., frame rate and resolution, (*ii*) how the energy consumption can be reduced from a systems perspective, and (*iii*) how the energy consumption can be reduced through image sensor hardware improvements. Our study includes fine-grained power measurement, modeling, prototyping, and model-driven simulation.

First, in Section 3, we report a detailed power characterization of five CMOS image sensors from two major vendors in the mobile market, breaking down the power consumption by major components and by operational modes. Based on the measurements and our understanding of image sensor internals, we construct power

MobiSys'13, June 25-28, 2013, Taipei, Taiwan
Copyright 2013 ACM 978-1-4503-1672-9/13/06 ...$15.00.

models that relate energy consumption to image quality requirements such as frame rate, resolution, and exposure time. By varying frame rate and resolution, we study the *energy proportionality* of image sensors; in particular, we consider how the energy cost for collecting a constant number of pixels changes when the frame rate and resolution changes. We observe that while power consumption decreases when frame rate or resolution drops, the energy per pixel increases significantly, up to 100 times more when reducing frame rate from 30 frames per second (FPS) to 1 FPS, which suggests poor energy proportionality. This observation suggests a key barrier in applying a well-known principle in energy-aware system design [6]: sacrifice quality (in this case, via frame rate and resolution reduction) for energy efficiency. Our characterization also reveals that the analog part of image sensors not only consumes a large portion of the power consumption (33-85% of sensor power) but also constitutes the bottleneck of energy proportionality.

Second, in Section 4, our investigation reveals two unexplored hardware mechanisms for improving energy proportionality: clock scaling and standby mode. Modern image sensors allow a wide range of external clock frequencies, but mobile systems often supply a clock of fixed frequency. We show that given the image requirement, there exists a frequency at which an image sensor consumes the lowest energy per pixel. Modern image sensors also provide a *standby mode* in which the entire image sensor is put into a non-functional, low-power mode. We show that standby mode can be applied between frames when the frame rate and resolution are sufficiently low. We call this optimization *aggressive standby*. We show that by combining clock scaling and aggressive standby, the energy proportionality of image sensing can be significantly improved, leading to almost constant energy per pixel across a wide range of image quality requirements and over 40% efficiency improvement when image quality requirement is low, e.g., one megapixel per frame and 5 FPS. In Section 5, we suggest several hardware modifications to further improve energy efficiency, in particular that of the analog parts.

Finally, in Section 6, using computer vision benchmarks and the data collected from the characterization, we demonstrate the quality vs. energy tradeoffs of image sensors with and without applying the optimizations described above. For continuous image registration on video, useful for image mosaicking and depth estimation, we can achieve a 36% power reduction by choosing an optimal clock frequency, and a 95% power reduction by using aggressive standby. Our suggested architectural modifications of image sensors can further reduce power. For example, by putting components in standby during exposure the power can be further reduced by 30%.

2 Background

We first provide an overview of the CMOS image sensor, the core of the camera on mobile systems. While cameras use optical and mechanical elements to focus light to the plane of the image sensor, we specifically discuss various electronic components and controls related to the image quality and power consumption after the light reaches the sensor.

2.1 Major Components of Image Sensor

A typical image sensor is a single chip that includes the following components as illustrated by Figure 1. The *pixel array* consists of an array of pixels; each pixel employs a photodetector and several transistors to convert light into charge stored in a capacitor. The *analog signal chain* employs active amplifiers and Analog-to-Digital-Converters (ADC) to convert the voltage of the capacitor into a digital output. Serial readout sensors employ a single analog signal chain for the sensor, while column-parallel readout sensors use one analog signal chain for each pixel column. The *image processor* performs basic digital image processing, such as demosaicking, denoising and white-balancing. The *I/O controller* interfaces the image sensor with the external world, usually the application processor in a mobile system. Along with streaming frame data, the I/O controller also receives instructions used to set the internal registers of the image sensor that determine the sensor's operational mode and parameters including frame rate and resolution. The *digital controller* manages the timed execution of the operations of the image sensor.

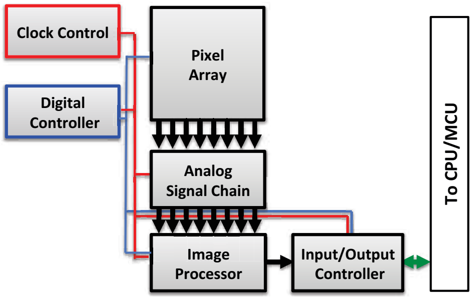


Figure 1: General image sensor architecture

2.2 Electronic Shutter (Exposure Control)

CMOS image sensors employ an *electronic shutter* to control the *exposure time*, T_{exp}, the length of time during which light can enter the sensor before a pixel capacitor is read out. Long exposures are used for low-light indoors scenes, while short exposures are used for bright outdoors scenes. There are two types of electronic shutters. (*i*) A *rolling shutter*, as shown in Figure 2, clears a row of pixels T_{exp} before it is to be read out. The rolling shutter then waits to clear the next row to prepare another row for exposure. The rolling nature allows the readout of some rows to overlap with exposure of other rows. However, with moving scenes, this causes temporal problems; although each row is exposed for a duration of T_{exp}, the top row of the frame is exposed much earlier than the bottom row of the sensor. (*ii*) A *global shutter* clears all rows of the pixel array simultaneously. After T_{exp} of exposure, the charge is transferred to a *shielded area*, a memory that maintains the state of the captured frame and frees the pixel array to subsequent exposure. As rows are read out from the shielded area, they do not face the moving effects suffered from rolling shutter operation. However, global shutters require memory for all pixels, and thus require expensive and complicated designs.

A programmable *shutter width* dictates the exposure time allotted by the electronic shutter. This allows systems developers to program the camera to operate in different ambient light environments. The shutter width is held as a register value and is implemented by the digital controller, which resets the charge of the pixel array capacitors appropriately.

2.3 Power, Clock, & Operational Modes

On mobile devices, the sensor is powered by multiple voltage rails, supplying the pixel array, the analog signal chain, the image processor, and the digital controller independently. We exploit these separate power rails to measure the power consumption of the various image sensor components and provide a characterization of the chip in Section 3.

An image sensor also uses an external *clock*. The clock controls the speed of the digital logic. Typically, an image sensor outputs

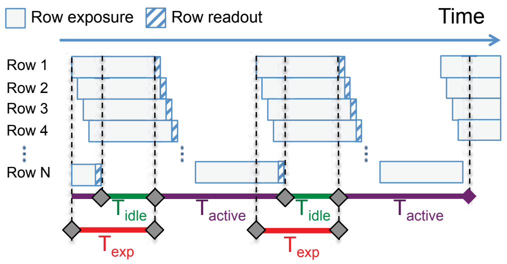


Figure 2: Streaming mode with rolling shutter

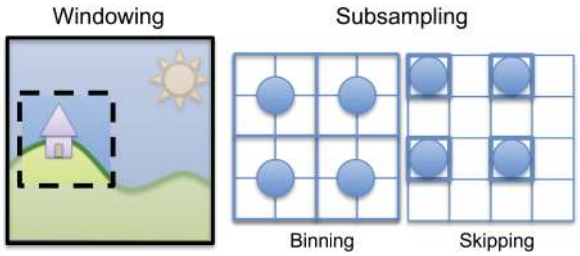


Figure 3: Image windowing and subsampling techniques

one pixel per clock period. Higher clock speeds allow sensors to process frames at different speeds, but consume significantly more power.

An image sensor typically provides two operational modes: streaming and standby. In *streaming* mode, the sensor alternates between two states: an idle state and an active state. During the *idle* state, the sensor is on and may be undergoing exposure, but the analog signal chain is not yet active to read out the pixel array. In the *active* state, the analog signal chain reads out the pixel array, the digital elements process the image and the I/O controller streams the frame out from the sensor. In Figure 2, the image sensor is in the streaming mode, alternating between T_{active} and T_{idle}. Because of the rolling shutter operation, T_{exp} can expose rows while rows are being read out during the T_{active} state.

In *standby* mode, much of the image sensor chip is put in a low-power mode with clock and/or power gated, but all register states are maintained, which allows for rapid wakeup. Standby mode consumes minimal power (0.5 - 1.5 mW). This mode is intended for taking snapshots where preview is not required; the sensor can remain in standby mode, wakeup to take a picture, and then return to standby.

2.4 Quality Controls

Typical image sensors provide controls to vary the quality of the frame, allowing for tradeoffs between frame resolution, field-of-view, frame rate, and power consumption. These are maintained by register values set through the I/O controller and controlled with the digital controller. We detail these operations below.

Frame rate R: The frame rate is the number of frames per second in the output stream. It is usually dictated by the system developer. The frame time, $T_{frame} = 1/R$, is the inverse of the frame rate. The minimum frame time is limited by the number of pixels in the image and the clock frequency. However, the frame time can be elongated by programming *Vertical Blanking*, which adds a number of “blank rows” to the image for timing purposes. Each blank row takes the same amount of time as reading a row out from the frame, but many components may be idle during the blanking time. The vertical blanking is manifested as rows of zeros in the image stream, and can be discarded by the processor receiving the output stream. Increased vertical blanking thus effectively raises the frame time, lowering the frame rate.

Frame resolution N: The frame resolution N indicates the number of pixels in the image, and directly influences the data transfer, processing, and storage requirements of the image sensor system. N can be reduced with two mechanisms: *windowing* and *subsampling*.*Windowing* directs the image sensor to output a smaller rectangular window of the frame, as shown in Figure 3. By specifying the size and location of the window, the system can request outputs with reduced fields-of-view. In contrast, *subsampling* preserves the field-of-view, but produces a “resized" lower resolution image. Image sensors use one of two techniques to achieve subsampling: (i) *Row/Column Skipping* skips sampling every other row or column of pixels. As a result, many pixels are not sent to the image processor, leading to rapid subsampled readout of an image. On the other hand, (ii) *Row/Column Binning* combines the values of adjacent pixels in the image processor after the analog signal chain. Groups of adjacent pixels create a single pixel value, reducing high-frequency aliasing effects and noise in the subsampled image. These techniques are shown in Figure 3.

2.5 Integration inside Mobile Systems

The image sensor is usually directly connected with the main application processor in a mobile device. Because large image sensors used in modern mobile devices require high data transfer speeds that cause synchronization issues on parallel buses, current devices use a serial interface between the image sensor and the application processor. For example, the Qualcomm Snapdragon S4 and Nvidia Tegra 3 use serial MIPI interfaces that consist of a clock to transfer data, one or more serial data paths, and a serial control bus [19].

Due to lack of hardware access, user applications on mobile devices resort to using the camera APIs provided by the operating system. The typical actions include control (e.g., focus the camera), image and video capture, and configuration (e.g., set resolution) of the camera. For example, the Windows Phone 8 native API provides `StartRecordingToSinkAsync()` for capturing an image and `StartRecordingToStreamAsync()` for recording a video, while the `AudioVideoCaptureDevice` maintains properties such as autofocus regions and exposure time. Control over frame rate and subsampling (but not windowing) parameters are also provided. Android and iOS SDKs provide similar APIs.

3 Energy Characterization

In this section, we report a characterization study of the energy consumption of several state-of-the-art CMOS image sensors. In particular, we evaluate the *energy per pixel* under various image quality requirements in terms of frame rate and resolution, which are relevant to computer vision applications. We have three objectives. First, we want a thorough understanding of how image sensors consume power in their major components. Second, we want to identify effective mechanisms to achieve the same quality with the lowest energy per pixel. And finally, we want to identify problems in the energy proportionality of existing and emerging image sensors: why does the energy-per-pixel increase as quality requirements decrease?

3.1 Apparatus and Image Sensors

We use a National Instruments USB-6212 16-Bit, 400 kilosample/second DAQ device for power measurements. We characterize five image sensors from two major vendors of CMOS image

Table 1: Important notations

Symbol	Description	Model (Source)
R	Framerate	
N	Number of pixels in a frame	
f	Clock frequency	
T_{frame}	Frame time	$T_{frame} = 1/R$
T_{active}	Time in active state	$T_{active} \approx N/f$
T_{idle}	Time in idle state	$T_{idle} = T_{frame} - T_{active}$
P_{idle}	Power consumption in idle state	$P_{idle} = a_1 \cdot f + a_2$ (Equation 10)
P_{active}	Power consumption in active state	$P_{active} = (b_1 \cdot N + b_2) \cdot f + b_3$ (Equation 12)
E_{frame}	Energy per frame	$E_{frame} = P_{idle}T_{idle} + P_{active}T_{active}$ (Equation 1)
P_{seq}	Power consumption for sequential frame capturing	$P_{seq} = \frac{P_{idle} \cdot (T_{frame} - T_{active}) + P_{active} \cdot T_{active}}{T_{frame}}$ (Equation 4)

sensors for the mobile market, as summarized by Table 2. By concurrently measuring the current into various voltage rails we are able to infer the power characteristics of the internal components of modern image sensors.

Table 2: Image sensors characterized in our study and power consumption at 24 MHz

	Max. Res.	P_{active}	P_{idle}	Market
A1	2592x1944	163.5 mW	161.9 mW	Snapshot
A2	768x506	189.5 mW	141.8 mW	Automotive
B1	3264x2448	338.6 mW	225.4 mW	Mobile
B2	2592x1944	225.1 mW	218.6 mW	Mobile
B3	752x480	137.1 mW	105.9 mW	Security

3.2 Breakdown by Components

We next provide our measurement results regarding the power consumption of the image sensors in idle and active modes, i.e., P_{idle} and P_{active}, and their breakdown into major components.

P_{active} *Breakdown*: We find that in the active state, the analog read-out circuitry consumes 70-85% of the total power, except for in B3, where it consumes only 33%, due to the column-parallel readout of its analog signal chain. The digital controller and image processing consumes 5%. The I/O controller that manages external communication consumes 10-15%. The breakdowns are shown for each sensor in Figure 4. As the bulk of the power is consumed by the analog signal chain, due to numerous power-hungry ADCs, this provides the greatest opportunity for new power-saving techniques, which we explore in Section 5.

P_{idle} *Breakdown*: Between frame captures, the sensors enter the idle state, where they still consume considerable power. The analog signal chain and image processor are powered during the idle state, but do not actively process pixels. In addition, I/O chains typically remain active during the idle state in order to communicate with the sensor to output blank rows or wait for register changes. As a result, the power of many components is typically reduced during the idle state. However, the amount of disparity depends on the image sensor architecture. For A2, B1 and B3, the analog power drops 15-45%. For A1 and B2, the analog components reduce their power minimally, less than 1%. The digital components for all of the sensors drop 10-55% and 3% for A1 and B2. For B2, the I/O power drops 40% and for A1 the I/O power drops 8%.

3.3 Energy Consumption Per Frame

We next examine the energy consumption per frame. Modern image sensors are programmed to capture a single frame (single shot) or to capture sequentially (video). For sequential frame capture, energy consumption per frame can be equivalently evaluated by the average power consumption in tandem with the frame rate.

In both cases, the energy consumption per frame depends on the power consumption of the operational modes and how much time the sensor spends in each mode. That is,

$$E_{frame} = P_{idle}T_{idle} + P_{active}T_{active} \tag{1}$$

From measurements and data sheets, we find that T_{active} is determined by the clock frequency, as one pixel is read out for every clock period. As the readout is pipelined with the digital processing and output of the image sensor, we can estimate:

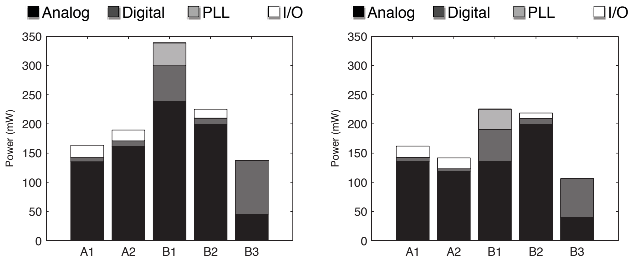
$$T_{active} \approx N/f \tag{2}$$

The idle time T_{idle} is determined by the exposure time for single frame captures and the frame rate for sequential frame captures. Figure 6 shows the power traces measured from the power rails of all of the sensors under sequential capture. The typical power consumption waveform clearly shows that the sensor alternately undergoes the active and idle states.

Single Frame Capture: For capturing a single photo, we care about the energy consumption for capturing a frame, E_{frame}^{single}. Figure 5(a) shows the power behavior of capturing a single image. The sensor must undergo exposure for T_{exp}, which ranges from 0.1 ms to 70 ms, depending on the lighting environment of the scene and the aperture size of the camera system (f/2.8 for typical smartphone cameras). The frame is then read out during T_{active}, after which the sensor may turn off. Thus, the energy consumption of a single frame capture can be simply modeled by inserting $T_{idle} = T_{exp}$ into Equation (1):

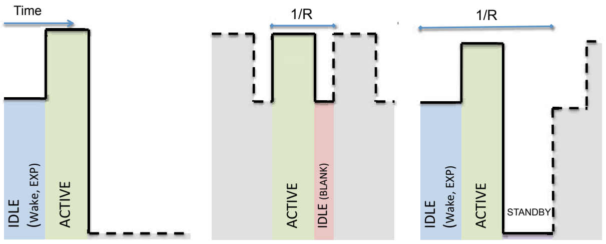
$$E_{frame}^{single} = P_{idle}T_{exp} + P_{active}T_{active} \tag{3}$$

Sequential Frame Capture: For sequentially capturing images, such as for video, we care about the average power consumption, P_{seq}. Figure 5(b) shows the power behavior of capturing sequential



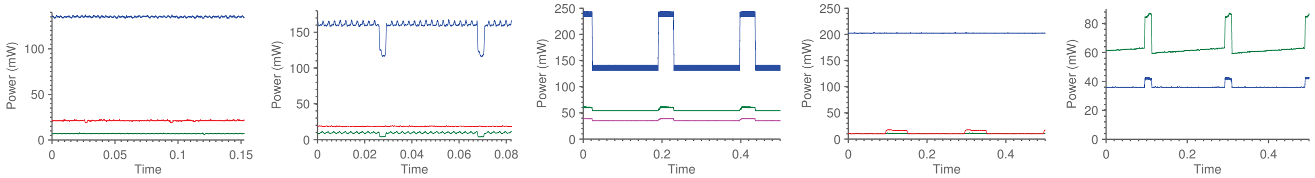
(a) P_{active} (b) P_{idle}

Figure 4: Average power of various rails in active state (a) and idle state (b), at 24 MHz



(a) (b) (c)

Figure 5: Power behavior for single capture (a), standard sequential capture (b) and sequential capture with aggressive standby mode (c)



(a) A1 (1 MP, 15 FPS) (b) A2 (1 MP, 20 FPS) (c) B1 (1 MP, 5 FPS) (d) B2 (1 MP, 5 FPS) (e) B3 (0.3 MP, 5 FPS)

Figure 6: Power waveform of image sensors. Analog (blue), digital (green), and I/O (red) voltage rails. For (c), the magenta line is the PLL voltage rail.

frames at a frame rate of R. Exposure can occur in either the active and idle states but because the exposure itself does not consume much power, this does not affect the overall power consumption. A cycle of capturing a frame can be clearly broken into two parts: the active state and the idle state, i.e., $T_{frame} = T_{idle} + T_{active}$. When the frame rate R is low, T_{idle} can be significant. The average power of sequential frame capture can be modeled as follows:

$$P_{seq} = \frac{P_{idle} \cdot (T_{frame} - T_{active}) + P_{active} \cdot T_{active}}{T_{frame}} \tag{4}$$

3.4 Energy Proportionality

In this section, we explore the energy implications of varying the quality parameters of frame capture. In particular, we vary the frame rate and resolution of the frame capture, model the power implications, and perform measurements for verification. Our measurements indicate that current image sensors are not energy proportional, as the energy consumption per pixel increases as the quality requirement decreases.

3.4.1 Frame rate

With a fixed clock frequency, the maximum frame rate of the sensor is the inverse of T_{active}. However, as explained in Section 2.4, the frame rate can be reduced by inserting blanking time. Then, for a given frame rate R, the energy per frame is:

$$\begin{aligned} E^{seq}_{frame}(R) &= P_{idle}(1/R - T_{active}) + P_{active}T_{active} \\ &= P_{idle}/R + (P_{active} - P_{idle})T_{active} \end{aligned} \tag{5}$$

Thus, we expect the energy per frame to increase as the frame rate decreases, as the power consumption becomes dominated by the idle power consumption. This is shown in Figure 7 by inserting measured P_{active} and P_{idle} values into the above equation. For each of the sensors, as the framerate drops from 20 FPS to 1 FPS, the energy per frame increases by an order of magnitude. Thus, image sensors are not energy proportional to frame rate. Instead, their energy per pixel increases as the performance requirement in terms of frame rate drops. In Section 4, we will show how the energy proportionality can be significantly improved by aggressively applying a power-saving standby mode during the idle state.

3.4.2 Resolution

When changing the resolution of the frame through subsampling or windowing techniques, fewer pixels are read out. Equation 2 indicates that T_{active} is proportional to the number of pixels and so a lower resolution will result in a shorter active time. Conversely, our measurements indicate that P_{active} and P_{idle} are only minimally influenced by the number of pixels, and thus remain unchanged for the purposes of our model. Then, we can model the energy for a single frame capture by plugging the numbers into Equation 1:

$$E^{single}_{frame}(N) = P_{active}N/f + P_{idle}T_{exp} \tag{6}$$

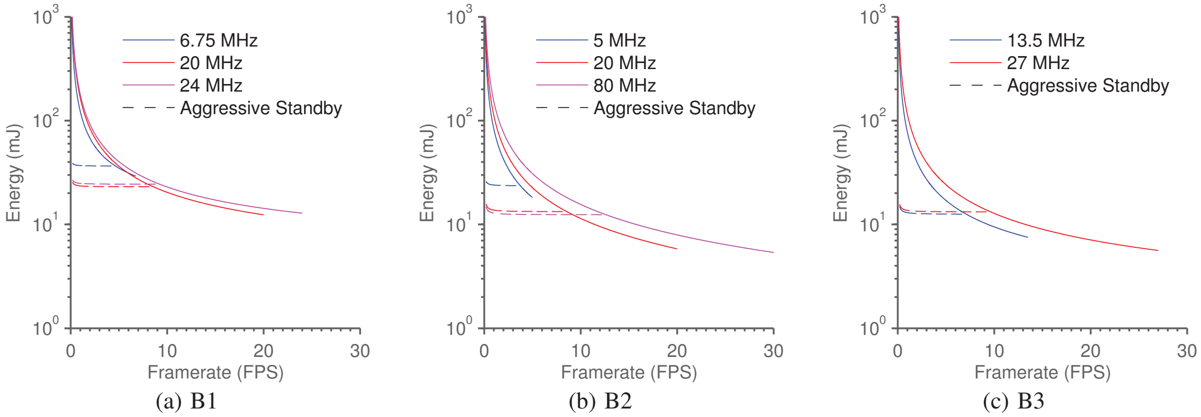
$$E^{single}_{frame}(N)/N = P_{active}/f + \frac{P_{idle}T_{exp}}{N} \tag{7}$$

For small T_{exp}, the second term is negligible. In this case, the energy per frame is reduced proportionally to N, as shown in Figure 8, and the energy per megapixel is nearly constant, as shown in Figure 9. Among sensors A1, A2, B1, and B2, the energy per megapixel is around 6 - 8 mJ/MP. B3 consumes lower energy per megapixel (3 mJ/MP), due to the low-analog-power nature of its column-parallel readout.

For sequential frame capture at constant frame rate, a shorter T_{active} requires a longer T_{idle} to keep T_{frame} constant. Then, building on Equation 5, with R fixed, we can model the energy of a frame and energy per megapixel as:

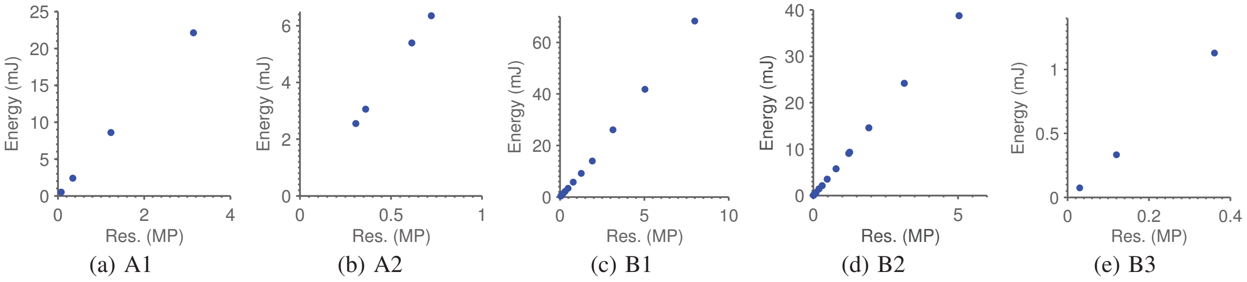
$$E^{seq}_{frame}(N) = (P_{active} - P_{idle})N/f + \frac{P_{idle}}{R} \tag{8}$$

$$E^{seq}_{frame}(N)/N = (P_{active} - P_{idle})/f + \frac{P_{idle}}{RN} \tag{9}$$



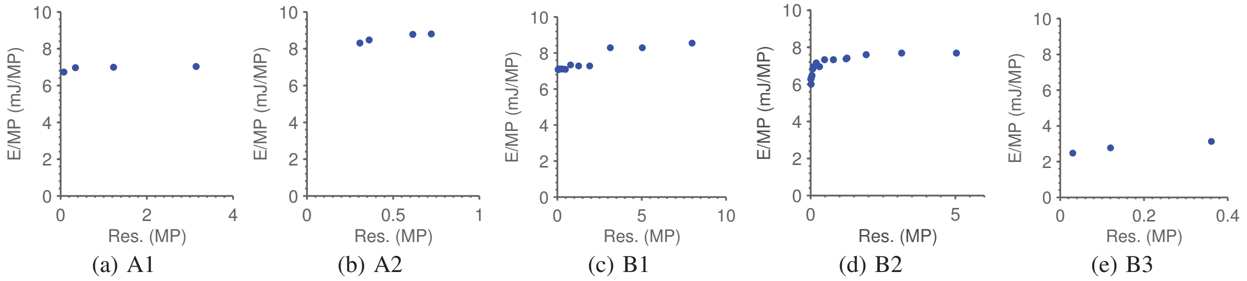
(a) B1 (b) B2 (c) B3

Figure 7: Modeled energy per frame in sequential frame capture without and with aggressive standby (1 MP frame)



(a) A1 (b) A2 (c) B1 (d) B2 (e) B3

Figure 8: Modeled energy per frame for subsampled single frame capture (with short T_{exp}, i.e., $E^{single}_{frame}(N) \approx P_{active}N/f$)



(a) A1 (b) A2 (c) B1 (d) B2 (e) B3

Figure 9: Measured energy per megapixel for subsampled single frame capture (with short T_{exp}, i.e., $E^{single}_{frame}(N)/N \approx P_{active}/f$)

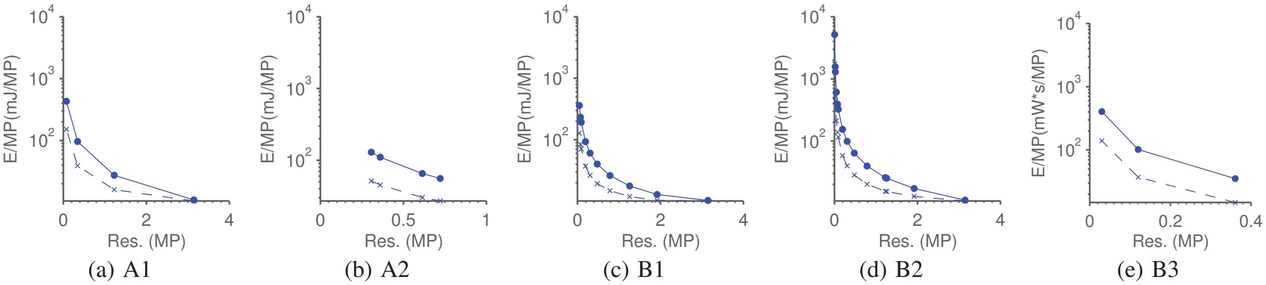
Given a constant frame rate R and a small resolution N, the energy per megapixel is dominated by the second term and is thus inversely related to the resolution of the subsampled frame, as shown in Figures 10 and 11, generated by simulating various framerate and resolution combinations with measured P_{active} and P_{idle} values. For example, for A1 at 1 FPS, the energy per megapixel rises by an order of magnitude as the resolution is dropped from 3 MP to 0.3 MP. Thus, as resolution is decreased, the energy per megapixel increases.

Our models and measurements indicate that current image sensors are not energy proportional to image quality reductions in framerate and resolution. In almost all cases, reducing the quality results in drastically higher energy per megapixel. The exception is the energy per megapixel of a subsampled single image capture, which remains relatively constant as resolution is decreased. In the next two sections, we explore existing mechanisms and propose future mechanisms to push towards energy proportionality.

4 Exploiting Existing Mechanisms

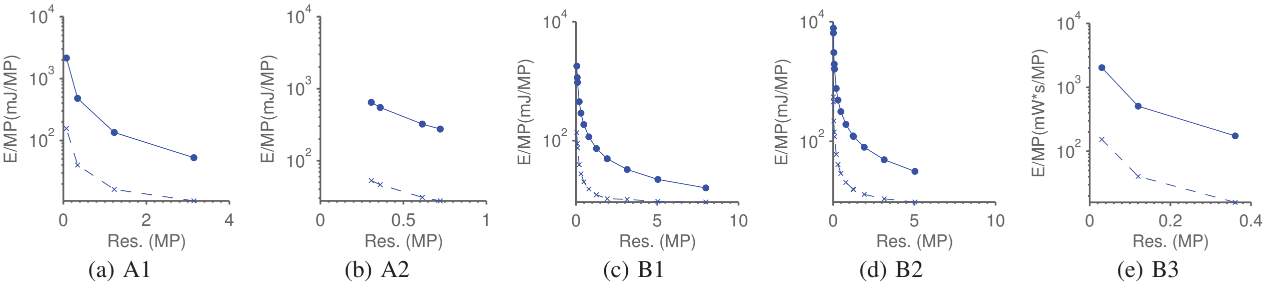
In this section, we exploit hardware mechanisms supported by modern CMOS image sensors to improve their energy efficiency. The key question we try to answer is: *given the frame rate (R) and resolution (N), what is the optimal configuration of an image sensor to achieve the lowest energy per frame?* The answer to this question can be implemented by the mobile system's image sensor driver to configure the sensor for energy efficiency when receiving requests from computer vision applications.

We identify two important existing power-saving mechanisms, *clock scaling* and *standby mode*, and answer the question by exploiting them. Modern mobile systems do not change the clock frequency of their image sensors nor do they apply standby mode to



(a) A1 (b) A2 (c) B1 (d) B2 (e) B3

Figure 10: Modeled energy per megapixel for subsampled sequential capture based on P_{active} and P_{idle} measurements at 5 FPS. Aggressive standby (from Section 4) is represented by the dashed line.



(a) A1 (b) A2 (c) B1 (d) B2 (e) B3

Figure 11: Modeled energy per megapixel for subsampled sequential capture based on P_{active} and P_{idle} measurements at 1 FPS. Aggressive standby (from Section 4) is represented by dashed line.

image capture because they intend the image sensors to be used for capturing high-resolution photo and fixed frame rate video, where clock scaling and standby mode bring little benefit. These mechanisms offer significant power efficiency when frame rate or resolution is low, which is sufficient for many computer vision tasks and for video streaming over networks. For 1 MP readouts, up to 50% of the power consumption of single frame capture and 30% of the power consumption of sequential frame capture can be eliminated by choosing the correct clock frequency. Further, by aggressively applying standby between frame captures, one can largely remove the idle energy consumption, leading to significant average power reduction, e.g., 40% for B1 at 5 FPS at 24 MHz.

4.1 Clock Scaling

Modern mobile systems do not change the clock frequency (f) of their image sensors. However, since the clock is supplied externally, its change only requires simple additional hardware, such as a programmable oscillator. For our experiments, we used a DS1077 oscillator, programmable over I^{2}C, and connected it to the external clock input on the B1, B2, and B3 image sensors.

Changing the clock frequency has significant implications on the image sensor's efficiency. We employ measurements with our understanding of the image sensor internals to quantify the relationship between f and the power consumption of an image sensor.

Our measurements, as summarized by Figure 12, show that both P_{idle} and P_{active} increase with f almost linearly. This is not surprising, since increasing the clock frequency linearly increases the switching power consumption of the digital and I/O parts of the circuit. (The clock frequency does not affect the analog signal chain power consumption, as these largely consume static power.)

We have:

$$P_{idle} = a_1 \cdot f + a_2 \tag{10}$$

$$P_{active} = c_1 \cdot f + c_2 \tag{11}$$

Table 3 summarizes the power model parameters for B1 to B3 according to our power vs. clock frequency measurements. Based on our understanding of how the clock works internally, we can further relate P_{active} to N as:

$$P_{active} = (b_1 \cdot N + b_2) \cdot f + b_3 \tag{12}$$

$b_1 \cdot N \cdot f$ denotes the power consumption by the analog signal chain, which reads out N pixels in each cycle of the clock. $b_2 \cdot f$ denotes the switching power consumption by the rest of the sensor, driven by the clock.

We make a few important notes about the above power models. First, b_3 is equivalent to c_2 and denotes the static power consumption of the sensor, independent of the clock. Second, a_1, a_2, and c_2 are intrinsic to the sensor and are independent of the frame rate or resolution. In contrast, c_1 increases as the number of pixels increases. Third, we have $c_1 \gg a_1$ and $c_2 \geq a_2$ because the digital circuitry stops switching in the idle state and the analog circuitry, while not driven by the clock, does not do additional work in the idle state.

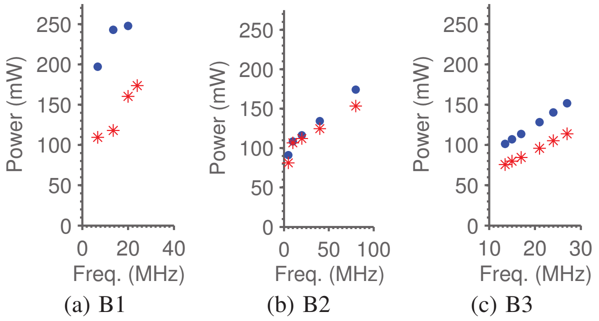
Using measurements and the models derived above, we next seek to answer the opening question by setting clock frequencies optimally.

4.1.1 Single Frame Capture

If we plug the models described above into the energy for a single frame capture, Equation 3, we can derive the energy consumption by single frame capture as:

$$E_{frame}^{single} = a_1 \cdot T_{exp} \cdot f + \frac{c_2 \cdot N}{f} + C \tag{13}$$

E_{frame}^{single} achieves the minimum when $f_{best}^{single} = (\frac{c_2 \cdot N}{a_1 \cdot T_{exp}})^{\frac{1}{2}}$.



(a) B1 (b) B2 (c) B3

Figure 12: Clock frequency f vs. P_{active} (blue dot) and P_{idle} (red stars)

Table 3 gives the f_{best}^{single} for B1-B3 under both indoor and outdoor exposure times and $N = 10^6$. Figure 13 also displays the energy for single frame captures for our measurements and the power model at different frequencies. As is evident from the table and the figure, the optimal frequency choice depends heavily on the exposure time. For outdoor usage, f_{best}^{single}, the optimal frequency choice, is typically higher than the sensor typically allows.

REMARK 1. *For single frame capture, the sensor's optimal clock frequency depends on the resolution (N) and exposure time (T_{exp}). For bright outdoors scenes, with short exposure times, the clock frequency should be set as fast as the sensor can handle.*

4.1.2 Sequential Capture

If we plug the frequency models above into our equation for sequential capture, Equation 4, we can derive the power consumption by sequential frame capture as:

$$P_{seq} = a_1 \cdot f + \frac{R \cdot N \cdot (c_2 - a_2)}{f} + B \qquad (14)$$

P_{seq} reaches its minimum when $f_{best}^{seq} = (\frac{R \cdot N \cdot (c_2 - a_2)}{a_1})^{\frac{1}{2}}$. Table 3 gives f_{best}^{seq} when $N = 10^6$ and $R = 5$ for B1 and B2. The optimal frequencies are within the range of clock frequencies allowed by the sensors. Therefore we have the following remark.

REMARK 2. *Without considering standby mode, the lowest power consumption for sequential frame capture can be achieved by carefully selecting the clock frequency depending on the frame rate (R) and the frame resolution (N).*

4.2 Aggressive Standby

We can also apply standby mode to idle time between two frames in sequential frame capturing as illustrated by Figure 5(c). During standby mode, the sensor consumes minimal power (e.g., 10 μW in standby mode vs. >100 mW in idle state). For simplicity, we ignore the wakeup time from standby mode, which occupies only tens of μs. The sensor performs no operation during standby mode, so a full T_{exp} cannot pipeline with the readout of the image pixels. As such, the duration of standby mode is $T_{standby} = T_{frame} - T_{exp} - T_{active}$. Therefore, we can calculate the average power consumption as

$$P_{seq}^{aggr} \approx \frac{P_{standby}(T_{frame} - T_{active} - T_{exp}) + P_{idle}T_{exp} + P_{active}T_{active}}{T_{frame}} \qquad (15)$$

Table 3: Parameters relating clock frequency f to power consumption. We assume 0.05ms and 50ms for T_{exp} outdoors and indoors, respectively. $N = 10^6$ and $R = 5$. All frequencies in MHz.

	B1	B2	B3
a_1	4.0E-06	8.2E-07	3.35E-06
a_2	76.2	90.1	4.4
c_1	5.6E-06	1.0E-06	5.1E-06
c_2	159.0	93.0	13.1
f_{best}^{single} (indoor)	28.2	47.6	19.0
f_{best}^{single} (outdoor)	564.4	951.9	379.2
$f_{best}^{seq}(5\ FPS)$	10.2	4.2	3.6

For clarity and simplicity, we ignore the standby power, i.e., $P_{standby} \approx 0$, since it is very small compared to P_{idle} and P_{active}. We have

$$P_{seq}^{aggr} \approx \frac{P_{idle}T_{exp} + P_{active}T_{active}}{T_{frame}} \qquad (16)$$

$$P_{seq}^{aggr} \approx a_1 \cdot R \cdot T_{exp} \cdot f + \frac{R \cdot c_2 \cdot N}{f} + D \qquad (17)$$

We note P_{seq} achieves its minimum when $f = f_{best}^{single} = (\frac{c_2 \cdot N}{a_1 \cdot T_{exp}})^{\frac{1}{2}}$. As we see above, the best frequency depends on the exposure time, given the quality requirement.

REMARK 3. *With aggressive standby, the sensor's optimal clock frequency for sequential frame capture depends on the resolution (N) and exposure time (T_{exp}). For bright outdoors scenes, with short exposure times, the clock frequency should be set as fast as the sensor can handle.*

We also note that in aggressive standby mode with a fixed clock rate and resolution size, the energy per frame remains constant as frame rate changes, as shown in Figure 7. This is due to the fact that frame rate is changed by extending the standby time, where the sensor consumes minimal power.

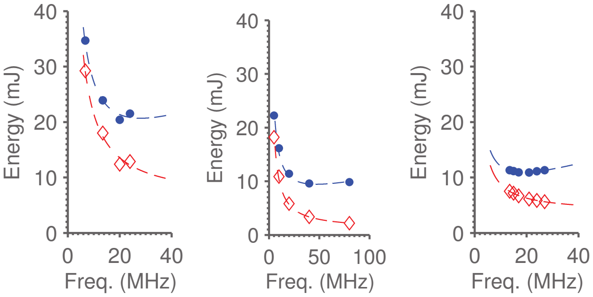
Hence, significant power reductions can result from application of clock scaling and aggressive standby. In our measurements, choosing an optimal clock frequency can reduce the power consumption of single frame capture by up to 50%. An optimal clock frequency can also reduce the power consumption of sequential frame capture by up to 30%. Additionally, by applying standby aggressively between frames, one can further reduce power consumption, e.g., 40% for B1 at 5 FPS at 24MHz.

5 New Power-Saving Mechanisms

Based on our findings, we next discuss a number of hardware modifications that further improve the energy efficiency of image sensors. Since the analog signal chain is the dominant power consumer in both idle and standby states and analog circuits are known to improve much slower than their digital counterparts, we focus on improving the efficiency of the analog signal chain without requiring a new design of analog circuitry.

5.1 Heterogenous Analog Signal Chains

Existing image sensors employ analog signal chains provisioned for the peak performance in terms of pixel per second supported by the image sensor. Because of this, while the pixel per second can be



(a) B1 (b) B2 (c) B3

Figure 13: Energy measurements of single frame capture at 1 MP with T_{exp}=50 ms (blue dot) and T_{exp}=0.125 ms (red diamond) at different f with theoretical models (dashed lines)

orders of magnitude lower in practice for continuous applications, the energy per pixel remains almost constant as shown in Figure 9. By using a much simpler analog signal chain for low performance capture, a much lower energy per pixel can be achieved in these situations.

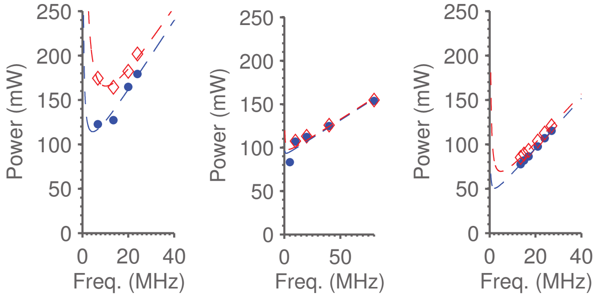
We suggest that an image sensor should include a heterogeneous collection of analog signal chains each optimized for certain bitrates. For example, one sophisticated chain could be active for full resolution, e.g., high-quality video taking, while another could be used when a much lower resolution is needed for computer vision applications. In both cases, the idle analog signal chain should be powered off.

To implement heterogeneous analog chains, extra but not duplicated circuitry is needed because the heterogeneous chains are not operational at the same time. For example, many complex modules of the analog signal chain, such as analog to digital converters (ADCs), will require only a small increase in hardware resources, since submodules of these modules can be shared between different implementations. For example, at lower resolutions, successive approximation (SAR) ADCs can be implemented by simple modifications to control logic to ignore least significant bits; similarly, lower resolution pipelined ADCs can be implemented by disabling the last pipeline stages that generate the least significant bits. Hence, image sensor designs with multiple analog chains require a careful balance between the increased cost due to extra hardware resources and the power savings achieved.

5.2 Fine-grained Power Management of Sensor Components

Existing image sensors provide a standby mode for the entire sensor. In Section 4.2, we showed how this mode can be aggressively applied to reduce the power consumption during the idle state. Now we explore the opportunity to apply power management (gating the power supply or clock) in a more fine-grained manner to reduce the power consumption during the active state.

Per Column Power Management of Analog Signal Chain: During readout, all column parallel analog signal subchains operate in parallel to read out a row of pixels simultaneously. However, during column skipping and windowing operations, not all pixels of a row need to be read out. The analog signal subchains for the skipped columns are left on in modern image sensors. As fewer pixels are addressed, these components should be shut off to save power. If only 1/2 of the columns are addressed, this would lead to substantial power savings, dropping the analog power by 50% and the total power by 30-40%.



(a) B1 (b) B2 (c) B3

Figure 14: Power for sequential capture of 1 MP frames at 1 FPS (blue dot) and 5 FPS (red diamond) at different f with theoretical models (dashed lines)

Power Management during Exposure: For single frame capture and sequential frame capture with aggressive standby applied, the power consumption during exposure time can contribute significantly to the total energy per frame or average power consumption, respectively. During the exposure time (T_{exp}), which can be long (e.g., 50 ms) under low illumination, most parts, including the digital components, the analog signal chain's amplifiers and ADC's, and the I/O, are in idle state, which still consumes substantial power. By putting these parts into the standby mode with either power or clock gated, the sensor would reduce the energy consumption of taking a single frame, i.e., Equation 3, and the power consumption of sequential capture, i.e., Equation 4. This has the effect of nullifying P_{idle}. It is easy to show that *when the power management is applied to the exposure time, the best clock frequency is always the highest possible regardless of the exposure time.* At this point, for long exposures, the sensor can consume fractions of the original energy cost of single frame capture; at T_{exp}=50 ms, B1, B2, and B3 would consume 19%, 83% and 50% less energy, respectively.

6 Energy Optimization for Continuous Vision Scenarios

Toward understanding the quality vs. efficiency tradeoffs possible for computer vision applications, we next specifically consider the power consumption of the image sensor during the execution of two fundamental computer vision problems: *image registration* and *object detection*.

Using the power models derived in Sections 3 and 4, we can model the image sensor power consumption when reducing the frame rate, reducing the window (field-of-view), and capturing the image at a lower resolution. In this section, we also apply the two power-saving mechanisms and gauge the impact on the performance of image registration and object detection. In doing so, we demonstrate that these mechanisms can reduce the energy consumption by 95% without sacrificing application performance. We also estimate the impact of suggested modifications from Section 5, reducing the energy consumption by 98% of the original.

Dataset

Our dataset consists of 90 seconds of 270x480, 30 FPS video from a smartphone mounted at chest level. The video was captured by a user walking around an outside path. We compute our machine vision tasks on adjacent pairs of frames of the video.

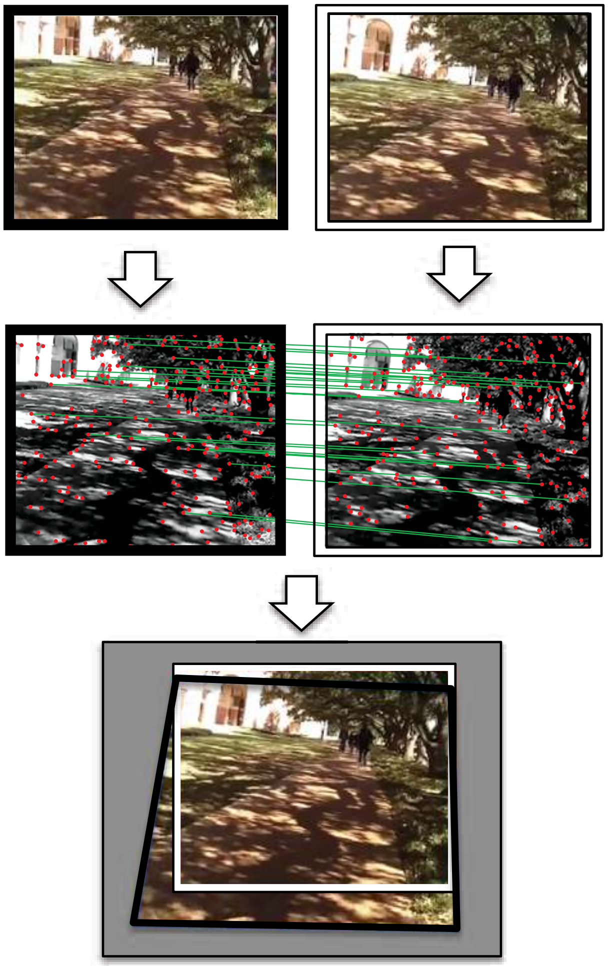


Figure 15: Image registration at 3 FPS. Corners (red dots) and homography inlier matches (green lines), along with image-mosaicked result.

To simulate low-resolution frame capture, we created image pyramids of the dataset by subsampling the resolution of the original frames. Each subsampled layer of the pyramid is constructed by taking the previous layer, convolving it with a Gaussian blur kernel, and removing multiples of rows. Subsampling by n is then defined by keeping every nth row. We also created windowed versions of our datasets: for a parameter W, we discard $W\%$ of the rows and $W\%$ of the columns from the borders of the image, effectively reducing the field of view. To simulate a reduction of frame rate to R FPS, we performed our vision tasks on pairs separated by $30/R$ frames.

6.1 Image Registration

Image registration – determining the correspondence points between two images – is a common problem in computer vision. Registration can be used to stitch images of a scene together, i.e., image mosaicking, to estimate the depth of objects, i.e., structure from motion, and to reduce shaking in video, i.e., image stabilization [8].

6.1.1 Algorithm

The registration algorithm involves finding corners in each image, matching corners in pairs of images, discarding outliers, and computing plane-to-plane transforms of the pair of images [7]. In this section, we describe the image operations necessary to compute the algorithm.

The Harris & Stephens corner detector [7] locates corners and edges in images by autocorrelating local patches around each pixel in an image. Where the autocorrelation value returns above a threshold, the algorithm detects a corner in the image. The patches around the corners in each image must then be matched with each other to generate correspondence pairs. This is done by correlating all possible pairs of corner patches. Where a corner in Image B is the maximum match of a corner in Image A and vice-versa, the pair of corners are determined to be a match.

With 4 or more corner matches, a plane-to-plane homography transform can be determined by fitting a 3x3 transform matrix to the set of corner pairs, e.g., using least squares. Because matches may be inaccurate, common homography algorithms use a Random Sample Consensus (RANSAC) to remove outliers from the list of matches. With a sufficient number of inliers, the homography is considered a success. In our implementation, we consider the existence of 25 inliers as the criterion for success.

6.1.2 Results

On our original dataset, the image registration process succeeded on 2783 frame pairs and failed on 7 pairs, for a success rate of 99.9%. Image registration also performs well with downscaled datasets. Frame rate reduction to 3 FPS still returned 95.7% success, 30% Windowing returned 96.5% success, and a downsampling to a resolution of 135x240 returned 91.8% success. Table 4 shows these quality parameters alongside their power consumption implications.

As shown in Figure 16, standard sequential capture does not significantly reduce the power consumption with lower quality requirements. However, by implementing clock scaling and aggressive standby modes, we can dramatically reduce the power by lowering the frame rate, window size, and subsampled resolution. For example, at 3 FPS, where image registration can still perform with 95.6% accuracy, the average power consumptions of B1, B2, and B3 are 185, 112, and 114 mW, respectively, using default configurations. By appropriately selecting the clock frequency, we can reduce the power consumptions to 106, 95, and 55 mW, giving a power savings of 36%. Aggressive standby further reduces the power consumptions to 9.9, 5.1, and 5.2 mW, or 5% of the original power consumption. Our proposed hardware modifications from Section 5 have significant power-impact when performing subsampling and windowing tasks, as columns of analog-signal chain are switched off. For W=30%, the modifications carry an estimated 75% reduction in power over aggressive standby mode, while for subsampling by 2, the modifications can reduce the power by an estimated 81%.

6.2 Object Detection

Detecting objects in frames is another fundamental and useful machine vision technique for understanding captured scenes. We apply the Viola-Jones Object Detection Framework [28], a widely-used platform for object detection, to detect the presence of human figures in our datasets.

6.2.1 Viola-Jones Object Detection Framework

The Viola-Jones framework detects objects in images based on their "Haar-like" rectangular features. A cascaded set of Adaboost-trained classifiers based on such features allows the framework to rapidly and robustly search image frames for objects from the library. While the original paper's example uses human faces as the subject, the

Table 4: Power consumption (in mW) for image registration (IR) success and person detection (PD) recall, for sequential capture P_{seq}, with optimal clock frequency $P_{seq}(f)$, with aggressive standby P_{aggr}, and with estimated architectural modifications P_{arch} for sensor B1

	IR Success %	**PD Recall %**	**N pix.**	P_{seq}	$P_{seq}(f)$	P_{aggr}	P_{arch}
Full Resolution	99.9%	94.4%	129600	202.2	154.2	99.1	32.7
Frame rate = 3 FPS	95.7%	83.3%	129600	185.8	106.1	9.91	3.27
Window, W = 30%	96.5%	77.8%	63504	192.9	129.5	71.6	17.8
Subsample by 2	91.8%	72.2%	32400	188.6	115.1	55.5	10.2

framework is robust to using other types of objects. We use it to detect human figures, using the `PeopleDetector` classifier from the Computer Vision Toolbox of MATLAB 2012b.

6.2.2 Results

Object Detection has fundamental challenges when objects in a scene are in unexpected poses or are occluded from view. However, in a continuous mobile vision scenario, the detection only needs to find an object once over all the frames in which the object is in view. Additionally, in such a continuous scenario, a preliminary detection at low quality could be followed by a high quality frame capture, which would check the validity of an object detection. Because we are primarily concerned with energy proportionality, we are most concerned with the low quality *recall* frame, ensuring that we detect an object when it is present in a scene.

To accommodate these relaxed expectations, we use a metric in which we count the number of false negatives on an instance basis rather than on a frame-by-frame basis. Then, our recall rate is (# of detected people)/(# of people).

Table 4 and Figure 16 shows the performance of Person Detection at various quality parameters on our 90-second dataset with 18 people in the scene. At full 270x480 resolution, Viola-Jones detects 17 of the people. As with image registration, scaling the frame rate offers the largest opportunity for energy proportionality, while still maintaining high performance. At 3 FPS, People Detection can detect 15 people, performing with 83.3% accuracy. The Viola-Jones performance weakens at lower resolutions, and low framerates reduce the chance that a person is detected. However, the balance between success rate and power offers computer vision developers the ability to carefully trade power for algorithmic performance, enabling low-power computer vision.

7 Related Work

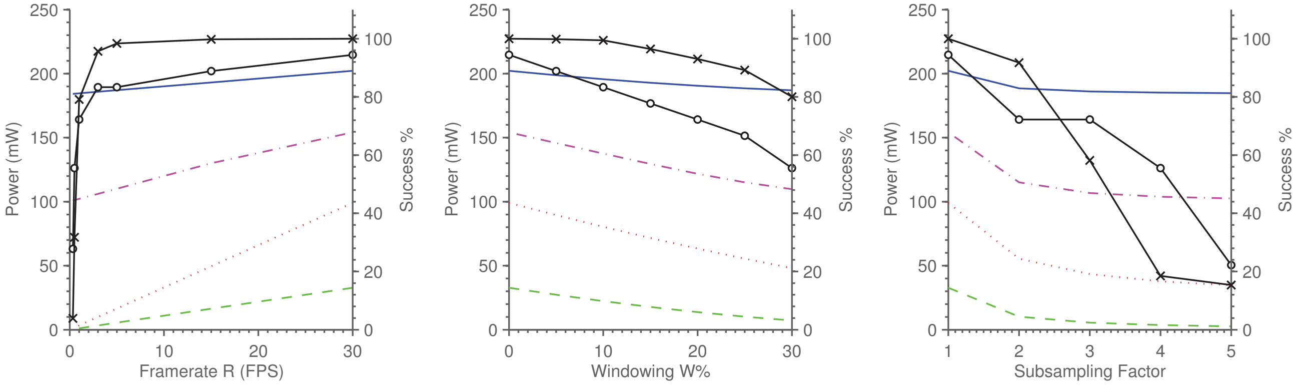
To the best of our knowledge, our work is the first publicly known study of the energy efficiency of image sensing from a system perspective. We next discuss related work in improving the energy consumption of cameras and image sensors.

CMOS Image Sensor Design: In this work, we study CMOS image sensors from a *system* perspective. We examine the power implications of sacrificing quality, which vision applications are likely to make, reveal inefficiency in the quality-power tradeoffs made by existing mobile image sensors, and suggest architectural modifications to improve the tradeoff. Our approach is complementary to that taken by the vibrant image sensor community, whose focus has been on improving image sensors through better circuit design. We refer the readers to textbooks on image sensor design for this approach, e.g., [22, 23]. It is well-known to image sensor designers that ADCs are often the power and performance bottleneck of high-speed, high-resolution image sensors, e.g., [3]. As the ADC is the interface between the physical and digital worlds in multiple domains, e.g., in sensors and wireless receivers, its performance and power efficiency has been extensively studied. We refer the readers to textbooks, e.g., [25] and survey papers, e.g., [13, 21] for recent development in ADC design.

Often, proposed techniques to address the ADC bottleneck involve many forms of compression, from temporal compression [17, 16, 11, 14, 4] to DCT [10] to predictive coding [15] to compressive sensing [26, 24]. These new architectures require significant modifications to the system and to camera applications. As a result, they are often intended for application-specific systems, e.g., surveillance camera networks [11]. In contrast, our presented techniques and modifications are evolutionary changes that can be easily incorporated into image sensors without any change to the system hardware designs or applications. Additionally, the goal of these sensor designs is orthogonal to ours: they target at reducing the power consumption of high-resolution capture, while we target at making the energy consumption proportional to image quality for efficient low-resolution capture.

Other Work Toward Efficient Vision Systems: Because image sensing is power-hungry, many have investigated the energy efficiency of a camera system at a high level for various platforms, but do not examine the internals of current image sensors for sources of inefficiency and mechanisms for software-based optimization as we do in this paper. Wireless visual sensor networks have tried both commercial-off-the-shelf image sensors and research prototypes like the ones discussed above [27] but are limited to much simpler applications like surveillance, due to an extremely tight power constraint. Many have made cameras wearable and a few have adventured to optimize the battery lifetime of the wearable cameras beyond simply duty cycling, e.g., [18, 9], and mobile phone designers are extraordinary careful not to quickly drain the battery, e.g., [1, 12]. The general approach has been to employ low-power sensors to manage the operations of the power-hungry image sensor. Without examining the internals of image sensors and their interface with the system and software, such work brings complementary benefits to our solutions.

We also note that power-saving mode and clock scaling have been extensively studied for microprocessors and digital circuits in general. Usually, clock scaling is combined with voltage scaling for maximal energy saving. For example, the authors of [20] show that given a processor, a workload and its deadline, there is an optimal way to apply clock/voltage scaling and power-saving mode jointly. For some processors, it is efficient to run as fast as possible and then enter a low-power mode, while for others, it can be most efficient to run as slow as possible. Our results in Section 4 show that image sensors have similar power-saving modes and allow clock scaling to reduce power consumption of the digital circuitry. Moreover, single and sequential frame captures can be considered as real-time workloads for image sensors. Image sen-



(a) Frame rate

(b) Windowing

(c) Subsampling

Figure 16: B1 Power consumption, image registration success (X) and person detection recall (O) at various quality parameters for sequential capture (blue solid), with optimal clock frequency (magenta dashed-dotted), with aggressive standby (red dotted), and with architectural modifications (green dashed)

sors, however, are distinct from microprocessors because the analog circuitry, which dominates power consumption, is not affected by clock or voltage scaling; a supply voltage change can ruin the accuracy of an analog-to-digital converter. Thus, power management and clock scaling solutions for microprocessors are not directly applicable to image sensors.

8 Discussion

System and API Support for Power-Saving Mechanisms: As we have demonstrated, clock scaling and aggressive standby power models described in Section 4 provide an opportunity to reduce the power consumption of the camera for tasks that do not require high frame rates and/or high resolutions. However, current mobile systems do not provide any system or API support for applications to adjust the clock frequency or apply standby mode. We hope our work will motivate platform and system vendors to consider such support.

Energy-Aware Computer Vision: Energy-proportional image capture opens the possibility of a new class of algorithms which carefully balance the tradeoff between accuracy metrics and power performance. With our power models, scaling the frame rate and resolution of an image has direct impact on the power consumption of a system. We plan to devise hierarchical/cascaded algorithms, using low-power image sensor modes to sense when to turn on progressively higher-power modes to detect information from a scene with low energy consumption. Other such algorithms from the vision community could leverage the quality-energy tradeoff to understand captured images and video on a continuous basis at low power.

9 Conclusion

Current image sensor operation is power-hungry and not energy-proportional. To explore this problem, we perform an experimental and analytical characterization of image and video capture on CMOS image sensors. We show two mechanisms for improving energy efficiency: *(i)* optimal clock scaling, which reduces power by up to 50% or 30% for one mega-pixel photos and videos, respectively; *(ii)* aggressive standby mode, which results in 40% power reduction for 1 MP, 5 FPS capture. We also suggest architectural modifications that further improve the energy efficiency of low-quality capture.

We use computer vision benchmarks to show application quality and energy efficiency tradeoffs that can be achieved with existing image sensors. For continuous image registration, a key primitive for image mosaicking and depth estimation, we achieve a 36% power reduction with an optimal clock frequency, and a 95% power reduction by using aggressive standby. Image sensor architectural modifications can further scale down the power consumption by an additional 30%. The quality-energy tradeoffs our work offers create new opportunities for continuous mobile vision under a power budget.

ACKNOWLEDGMENTS

The authors thank Eddie Reyes for his assistance in MATLAB programming for image registration and person detection benchmarks. The authors also thank the scientists at Aptina Imaging for corroborating our observations about the capabilities of modern day image sensor architectures and for answering our technical questions. The authors are grateful for the useful comments made by the anonymous reviewers and the paper shepherd, Dr. Shyamnath Gollakota. This work was supported in part by NSF Awards #0923479, #1012831, and #1054693.

REFERENCES

[1] M. Azizyan, I. Constandache, and R. Roy Choudhury. SurroundSense: mobile phone localization via ambience fingerprinting. In *Proc. ACM Int'l Conf. on Mobile Computing and Networking (MobiCom)*, 2009.

[2] P. Bahl, M. Philipose, and L. Zhong. Vision: cloud-powered sight for all: showing the cloud what you see. In *Proc. ACM Wrkshp. Mobile Cloud Computing and Services (MCS)*, pages 53–60, 2012.

[3] Y. Chae, J. Cheon, S. Lim, M. Kwon, K. Yoo, W. Jung, D.H. Lee, S. Ham, and G. Han. A 2.1 M pixels, 120 frame/s CMOS image sensor with column-parallel $\Delta\Sigma$ ADC architecture. *IEEE Journal of Solid-State Circuits*, 46(1):236–247, 2011.

[4] N. Cottini, L. Gasparini, M. De Nicola, N. Massari, and M. Gottardi. A CMOS ultra-low power vision sensor with image compression and embedded event-driven energy-management. *IEEE Journal of Emerging and Selected Topics in Circuits and Systems*, 1(3):299–307, 2011.

[5] A. El Gamal and H. Eltoukhy. Cmos image sensors. *IEEE Circuits and Devices Magazine*, 21(3):6–20, 2005.

[6] J. Flinn and M. Satyanarayanan. Energy-aware adaptation for mobile applications. *ACM SIGOPS Operating Systems Review*, 33(5):48–63, 1999.

[7] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50. Manchester, UK, 1988.

[8] Richard Hartley. *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge, UK New York, 2003.

[9] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood. SenseCam: A retrospective memory aid. In *Proc. ACM Int'l Conf. on Ubiquitous Computing (UbiComp)*, 2006.

[10] S. Kawahito, M. Yoshida, M. Sasaki, K. Umehara, D. Miyazaki, Y. Tadokoro, K. Murata, S. Doushou, and A. Matsuzawa. A CMOS image sensor with analog two-dimensional DCT-based compression circuits for one-chip cameras. *IEEE Journal of Solid-State Circuits*, 32(12):2030–2041, 1997.

[11] D. Kim, Z. Fu, J.H. Park, and E. Culurciello. A 1-mW CMOS temporal-difference AER sensor for wireless sensor networks. *IEEE Transactions on Electron Devices*, 56(11):2586–2593, 2009.

[12] N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A.T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 9, 2010.

[13] B. Le, T.W. Rondeau, J.H. Reed, and C.W. Bostian. Analog-to-digital converters. *IEEE Signal Processing Magazine*, 22(6):69–77, 2005.

[14] J.A. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco. A 3.6 s latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE Journal of Solid-State Circuits*, 46(6):1443, 2011.

[15] W.D. León-Salas, S. Balkir, K. Sayood, N. Schemm, and M.W. Hoffman. A CMOS imager with focal plane compression using predictive coding. *IEEE Journal of Solid-State Circuits*, 42(11):2555–2572, 2007.

[16] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128× 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008.

[17] U. Mallik, M. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings. Temporal change threshold detection imager. In *Digest of Technical Papers from IEEE Int. Solid-State Circuits Conference*, pages 362–603, 2005.

[18] WW Mayol, B. Tordoff, and DW Murray. Towards wearable active vision platforms. In *Proc. IEEE Int.Conf. Systems, Man, and Cybernetics*, 2000.

[19] MIPI camera interface specifications. http://www.mipi.org/specifications/camera-interface.

[20] A. Miyoshi, C. Lefurgy, E. Van Hensbergen, R. Rajamony, and R. Rajkumar. Critical power slope: understanding the runtime effects of frequency scaling. In *Proc. ACM Int'l. Conf. Supercomputing*, pages 35–44, 2002.

[21] B. Murmann. A/D converter trends: Power dissipation, scaling and digitally assisted architectures. In *Proc. IEEE. Custom Integrated Circuits Conference*, pages 105–112, 2008.

[22] J. Nakamura. *Image sensors and signal processing for digital still cameras*. CRC, 2005.

[23] J. Ohta. *Smart CMOS image sensors and applications*, volume 129. CRC, 2007.

[24] Y. Oike and A. El Gamal. CMOS image sensor with per-column $\Sigma\Delta$ ADC and programmable compressed sensing. *IEEE Journal of Solid-State Circuits*, 48(1), 2013.

[25] M.J.M. Pelgrom. *Analog-to-Digital Conversion*. Springer, 2010.

[26] R. Robucci, J.D. Gray, L.K. Chiu, J. Romberg, and P. Hasler. Compressive sensing on a CMOS separable-transform image sensor. *Proceedings of the IEEE*, 98(6):1089–1101, 2010.

[27] S. Soro and W. Heinzelman. A survey of visual sensor networks. *Advances in Multimedia*, 2009.

[28] Paul A. Viola and Michael J. Jones. Robust real-time face detection. In *Proc. Int'l. Conf. Computer Vision (ICCV)*, 2001.